GENERATING AWARENESS FROM COLLABORATIVE WORKING ENVIRONMENT USING SOCIAL DATA

Daniel Gallego Vico, Iván Martínez Toro and Joaquín Salvachúa Rodríguez
Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid
Avda. Complutense 30, Ciudad Universitaria, 28040 Madrid, Spain
{dgallego, imartinez, jsalvachua }@dit.upm.es

ABSTRACT

Nowadays, Internet is a place where social networks have reached an important impact in collaboration among people over the world in different ways. This paper proposes a new paradigm of building CSCW tools for business world following these new ideas provided by the social web to collaborate and generate awareness. An implementation of these concepts is described, including the components we provide to collaborate in workspaces, (such as videoconference, chat, desktop sharing, forums or temporal events), and the way we generate awareness from these complex social data structures. We also present figures and validation results in the paper to stress that this architecture has been defined to support awareness generation via joining current and future social data from business and social networks worlds, based on the idea of using social data stored in the cloud.

KEYWORDS

Awareness, Social networks, Collaboration patterns, CSCW, Web 2.0, ROA.

1. INTRODUCTION

In the last years, due to the Web 2.0 movement, the social web applications are one of the areas where new technologies are more important, not only to demonstrate the capacities of these new technologies, but also to socialize Internet. As a result of this evolution, different kinds of relevant social networks have appeared such as Twitter or Facebook. Users connect themselves with the entire world in the way they prefer, generating an incredible knowledge shared in the Internet and in some cases, accessible for everyone.

On the other hand, the business world have understood the advantages of this social way of collaboration, because one of the most important problems in today enterprises is keeping its employees "know how" in case they leave the company. Besides, revealing the collective activity and generating awareness from it, is also an essential concern that enterprises have. For this reason, the Computer Supported Cooperative Work (CSCW) tools have been significant for business world since they appeared in the 1980s (Grudin, 1994). These tools used to have a rigid structure completely different from the social web movement concepts. However, applications like Microsoft Office Communicator 2007 and IBM Lotus Connections are starting to evolve to more dynamic schemes.

In this paper, we propose a new point of view of CSCW tools according to social web ideas, in order to offer the common aspects of these tools (collaboration among users), and the possibility to generate social awareness from the data structures in which is based our application, so users can study their own collaborative context and their ties with other users. This new CSCW 2.0 that we have developed is named CollawareSoft. It has been deployed within a banking environment and in the last months it has been used to provide advanced collaboration in this business context.

The structure of the paper is divided as follows: the next section gives a summary of the most important user requirements, and a common case of use starred by an imaginary worker called Bruce is presented. Section 3 describes the architecture and the main concepts of the implementation of this application. Section 4 aims to place the main ideas behind Social Awareness. The following two sections describe the way awareness is generated from CollawareSoft, and the results obtained, solving the awareness Bruce's problem. Finally, the last section presents the conclusions from this implementation and further works.

2. FUNCTIONAL AND USER REQUIREMENTS

Before building CollawareSoft, we decided to analyze our users' requirements. Also, it was important to place the functional features that a CSCW system must include. For this reason, we are going to follow the taxonomy described by Reinhard et al (1994: 29), so as to achieve a solution to the first problem (keeping the "know how" in the company and allowing collaboration among workers):

- *Interaction*: our idea is to follow the social web movement, so synchronous and asynchronous interaction is available for users.
- *Coordination*: it is necessary to establish different user roles such as administrator, reviewer, publisher or invited.
- *Distribution*: we chose Client-Server architecture with a web client which helps users to interact in a distributed way.
- *User-specific reactions*: the system should react depending on the individual roles of each user, offering a different user interface based on his permissions.
- *Visualization*: we built a system that follows the Rich Internet Applications (RIAs) tendencies (Moritz, 2008), where the Human-Computer Interaction (HCI) is near to desktop applications and the usability is higher than in traditional web pages.
- *Data hiding*: CollawareSoft separates public from private data using workspaces or collaborative groups.

Arrived to this point, we can add a new important property focused on solving the second problem raised in the introduction:

- *Social awareness*: generating user context from social and temporal structures of collaboration helps people to understand the activities and projects around them. Besides, this awareness provides important information about the position of the user in the groups he collaborates. Hence, a good way to achieve this is by building contextualized awareness from CollawareSoft's social data.

Finally, to describe what we mean with the properties we have just set, we are going to see how our proposal can help to resolve a common problem in everyday collaboration:

"*Bruce is a software designer who is starting to work in a new department of his company, in a project related to cloud computing topics. He does not know much about these concepts. As a consequence, he needs to talk with his new co-workers to bring up to date himself, but he does not know who the experts are. What can he do?"*

At the end of the paper, we will see how our system resolves this current problem in today enterprises.

3. COLLABORATIVE PLATFORM: COLLAWARESOFT

In this section, we describe the general details of the implementation of the collaborative platform CollawareSoft, developed following the Scrum methodology, an agile process that can be used to manage and control complex software and product development using iterative and incremental practices.

The architecture chosen was Client-Server, in order to achieve the distribution property saw previously.

3.1 Server

3.1.1 SRI: Social Resources Infrastructure

It is the infrastructure responsible for providing the social data and the collaborative structures needed to build the system, and it acts as a gateway between the data base and the web client. The SRI is an application implemented in Ruby on Rails (Thomas and Heinemeier, 2006) with Model View Controller (Curry and Grace, 2008: 88-89) architecture. It also has a REST API (as we can see in Figure 1) designed following the concepts put forward by Fielding (2000). It provides communication with the web client, so the client can manage these social data stored in the data base through HTTP methods.

The REST resources offered by the SRI are: *users*, *spaces* (identify a group of users or a specific project), *events*, *articles* (forum posts and comments allowing asynchronous interaction) and *performances* (specifies the different roles that a user has in the spaces he belongs to, providing coordination).

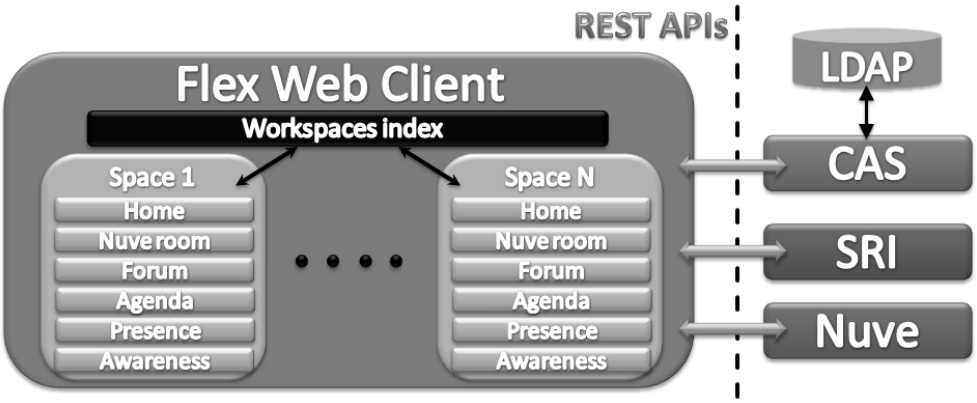

Figure 1: CollawareSoft architecture where the web client communicates with server modules through REST APIs

In addition to this, we must emphasize that the format chosen to communicate information between the SRI and the web client was The Atom Syndication Format (RFC 4287, 2005) to read the resources, and The Atom Publishing Protocol (RFC 5023, 2007) to create and update the resources. This allows CollawareSoft to connect with other social applications, such as Google and Blogger which also use Atom.

3.1.2 Nuve: Videoconference as a Service

As we can see in Figure 1, another module in the server side is Nuve (Rodriguez et al, 2009). It offers videoconference rooms as a service in a cloud computing way. It also offers users access to a collaborative synchronous interaction environment including audio, video, chat and shared applications.

Nuve, developed by the authors' research group, is the evolution of Marte 3.0 (Cerviño et al, 2008), a videoconference Client-Server service. The Nuve architecture extends the past implementation to a scalable Cloud Computing service which provides virtual rooms to users based on the CollawareSoft demand.

Furthermore, when we create a new space, it has associated a Nuve room where the space users are registered. In consequence, we have a one by one relationship between SRI spaces and Nuve rooms.

3.1.3 Authentication: CAS and LDAP

We use a CAS module to have a Single-Sing-On service to authenticate the web client in the SRI. The client communicates with this module through the RESTful API (Jasig Wiki, 2009), and the CAS module interacts with an LDAP database that stores the users and their performances.

3.2. Client

The CollawareSoft web client has been developed following a RIA paradigm, using Adobe Flex for its implementation, covering user-specific reactions and visualization requirements. This has an additional benefit: the user does not have to install any application in most cases because most browsers today have the Flash Player plug-in installed. In particular, we use the open source framework Cairngorm (Adobe Open Source, 2007) to design the application because it provides Model View Controller (MVC) architecture. As a result, the client is principally based on a collaborative structure where the space is the main unit in the application. In other words, users belong to one or more spaces where they collaborate among themselves using a set of collaborative tools created via the SRI's social data.

Thus, every space has the following tools, available for all the users registered (illustrated in Figure 1): *Home* (informs about the next space events and the recent posts in the forum), *Nuve room* (provides a videoconference room with chat and desktop sharing), *Forum* (based on the SRI's articles resource), *Agenda and Calendar* (based on the SRI's events resource, meetings and events can be planed), *Profile and Presence* (provides the users' public profile information in the space and a presence service to know who is on-line) and *Social Awareness* (generates collaborative context for a user or a group of users in a space, so they can be aware of the collaborative life that exists in the application).

Besides, due to the fact that we have a single client that integrates all the server modules, the orchestration among the services provided by them is accomplished.

4. WHAT IS SOCIAL AWARENESS?

Before talking about awareness itself, we must first take into account the concepts behind the awareness generation. In our case, we follow the ideas described by Fisher and Dourish (2004):

- *Social Networks*: they are probably the most important part of the awareness generation. The Social Network analysis has been broadly studied in the social science area (Wasserman and Faust, 1994). They describe the relationships that exist in sets of people by analyzing some social aspects or personal links among them. Also, using social networks we can find out social structures and working groups, understanding special roles of some of their members.
- *Temporal Structures*: they describe how the social networks evolve through time. Thus, they show the collaborative rhythms that a specific person experiments in the different groups or projects he belongs to.

As a consequence, the joint of these two concepts and their analysis generates the social awareness that we are looking for.

5. GENERATING AWARENESS FROM COLLAWARESOFT

Now that we have reached this point, the question is obvious: how do we generate collaborative context on CollawareSoft? We have a very wide and rich set of social data structures which can be used as an input to generate collaborative patterns so we are going to raise the different options to generate this information using the resources provided by the SRI: *users*, *spaces*, *articles* and *events*.

First of all, we must take into account that a space is a collaborative structure itself. Thus, the first thing a user notices when he enters in a space is the users registered in it. As a consequence, the generation of collaborative patterns based on the *spaces* and *users* resources is immediate.

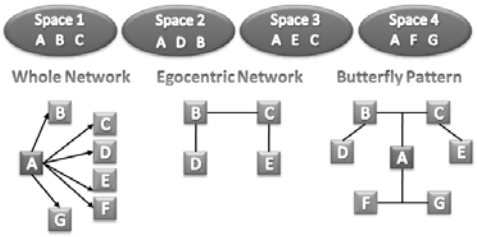

Figure 2: Generation of collaboration patterns from spaces and users

Figure 2 shows an easy example about how the system internally generates awareness. As we can see, user A belongs to four spaces in which he collaborates with several users. Regarding the Whole Network pattern, we can get what are his collaboration issues with these users, by checking over the users that share spaces with him. If we attend to the Egocentric Network (Fisher, 2005) pattern, we get information about the awareness relating different users to each other. According to the analysis of the users that share spaces with A, we can notice that the work teams formed by (D, B) and (E, C), that seemed to be disjoint, are in fact linked due to the collaboration between A with B and C in space 1. Then, we can see in the scheme how the relation is so, because of these users. Finally, the Butterfly pattern or Dual Roles pattern informs us that A is the union nexus between two different working groups and that is why we can use these patterns to identify the A´s role inside the company and locate him in its collaborative context in a more accurate way.

On the other hand, by analyzing forum conversation threads, we can extract social information. Thus, we can connect users among themselves regarding the parent-child (post-comment) structure of these messages. Furthermore, we add time analysis to provide temporal structures to the social networks we had generated before. We can do this because all the posts and comments published have a date and an author that becomes into messages more or less significant.

Another way of generating awareness is by using the joint of *events* and *users* resources from the Agenda and Calendar component, attending to the topics and users groups that created these events in the space.

Finally, there is another way of analyzing in a higher level these social data. If we take into account that usually, the collaborative patterns do not appear isolated and maybe they could overlap, we can combine them to generate *Macro Patterns*.

This kind on new patterns will give us a more complete and truthful information about the awareness of a user or a group than simple patterns. For example, the Butterfly pattern illustrated in figure 2 has implicit inside it the Egocentric Network pattern. Therefore, this new macro pattern shows us more information about the awareness of the user A. Consequently, when we want to generate awareness, we can analyze our social data in several levels depending on the detail and user interrelationship level that we are looking for.

6. RESULTS

Now is the time to show some results from our work. To do this, we are going to solve the awareness problem that we raised at the beginning of this paper with Bruce's case. We are going to suppose that the company where Bruce is working has deployed CollawareSoft to provide collaboration among their workers. In this case, "*Bruce uses the Experts search engine present in the application to look for cloud computing experts in his new working project which has a space (named DIT) in the application*".

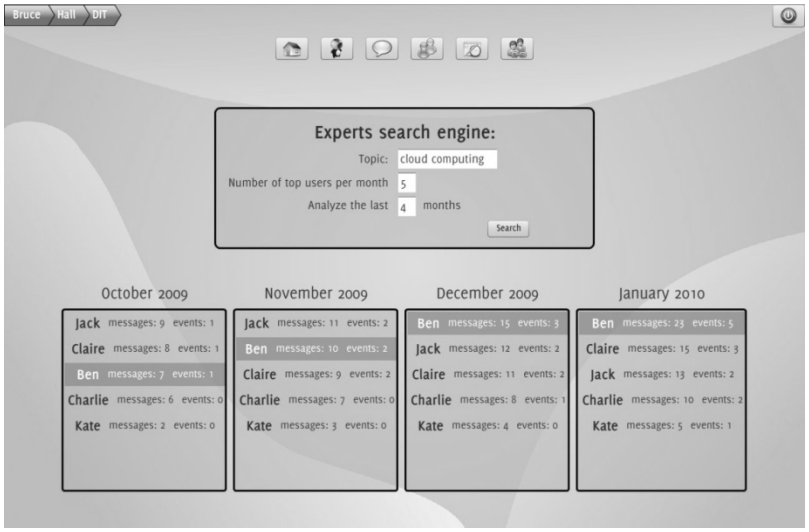

Figure 3: A CollawareSoft view of the social awareness generation component

The figure 3 illustrates that the result is a Top 5 macro pattern that shows the cloud computing experts in the space during the last four months. Our awareness generation tool has internally analyzed the temporal evolution of the users that have written messages in the forum or has organized events about these topics. Then, it has ordered them attending to its importance, taking into account the social ties among themselves.

As a result, the tool remarks the most important expert due to his number of messages and events, besides his evolution through the months analyzed. "*Now Bruce knows who the experts he was looking for are, and in particular, he knows that Ben is the greatest expert of his new department in the cloud computing area. Therefore, once this awareness has been generated for Bruce, he could now talk to Ben in real time (using chat or videoconference), follow his conversation threads in the forum or join the next events organized by Ben in the Agenda and Calendar component*".

7. CONCLUSIONS AND FURTHER WORKS

We have built a web application called CollawareSoft capable of providing collaboration at different levels, and also awareness generation to provide enterprise's users a more complete social and collaborative experience following the RIA paradigm. Therefore, throughout this paper we have shown the main features of the Client-Server architecture built, and the different ways we have to generate awareness from the social data resources we stored in the server side. Afterwards, we have demonstrated that problems such as looking for knowledge in a company can be solved. Hence, this tool is capable of analyzing social data to create social and temporal patterns that we can combine in order to achieve complex pattern (or macro patterns) that

give us rich awareness. The evaluation method employed to validate this tool is similar to the one used by Gutwin et al (2004). We can also talk now about future lines of investigation that we have identified.

Firstly, discovering new methods and algorithms to analyze relationships among users and social data are important areas where we should work in the future in order to include them in our application.

Secondly, regarding the ideas proposed by the Social Web Incubator Group of the W3C, specially the Socially Aware Cloud Storage concept (Berners-Lee, 2009), that tries to achieve a way to re-architect social network applications (e.g. Facebook) so as to use their user's data by third parties, we have an interesting research area. If we think about the present Social Network Site Silos where every social network has its own user's data only shared through specific APIs, we quickly understand that it is necessary a way to achieve a common architecture to share these user's data, allowing third parties to use them in a cloud computing way.

Thus, we believe that if this kind of architecture is achieved, we could use it in the future to join the social data that a user has in these social networks with the data that is allocated in CollawareSoft, achieving a more complex and rich awareness of this user, to provide him a better collaborative and social context. So, to conclude, we assert that joining the user's social data from the personal and business world will probably give us, in the future, a powerful way to generate awareness.

ACKNOWLEDGEMENT

This work has been supported by the ITECBAN project, sponsored by the CDTI and the Spanish Ministry of Industry, Tourism and Commerce. The authors would like to express their gratitude to INDRA Sistemas S.A.

REFERENCES

Adobe Open Source, 2007. Cairngorm. [Online], Available: http://opensource.adobe.com/wiki/display/cairngorm/Cairngorm [10 March 2010].

Berners-Lee, T., 2009. Socially Aware Cloud Storage. First draft. [Online], Available: http://www.w3.org/DesignIssues/CloudStorage.html [5 March 2010].

Cerviño, J. et al, 2008. Marte 3.0: una videoconferencia 2.0. *Proceedings of JITEL 2008*. Madrid, Spain, pp. 209-216.

Curry, E. and Grace, P., 2008. Flexible Self-Management Using the Model-View-Controller Pattern. *IEEE Software*, Vol. 25, No. 3, pp. 84-90.

Fielding, R., 2000. Architectural Styles and the Design of Network-based Software Architectures. *Doctoral dissertation*. University of California, Irvine, USA.

Fisher, D., 2005. Using Egocentric Networks to Understand Communication. *IEEE Internet Computing*, Vol. 9, No. 5, pp. 20-28.

Fisher, D. and Dourish, P., 2004. Social and temporal structures in everyday collaboration. *Proceedings of Conference on Human Factors in Computing Systems*. Vienna, Austria, pp. 551-558.

Gutwin, C., Penner, R. and Schneider, K., 2004. Group awareness in distributed software development. *Proceedings of ACM Conference on Computer Supported Cooperative Work*. New York, Chicago, Illinois, USA, pp. 72-81.

Grudin, J., 1994. Computer-supported cooperative work: history and focus. *Computer*, Vol. 27, No. 5, pp. 19-26.

Jasig Wiki, 2009. RESTful API CAS User Manual. [Online], Available: http://www.ja-sig.org/wiki/display/CASUM/RESTful+API [9 March 2010].

Moritz, F., 2008. Rich Internet Applications (RIA): A Convergence of User Interface Paradigms of Web and Desktop - Exemplified by JavaFX. *Diploma Thesis of University of Applied Science Kaiserslautern*. Zweibrücken, Germany.

Reinhard, W. et al, 1994. CSCW tools: concepts and architectures. *Computer*, Vol. 27, No. 5, pp. 28-36.

RFC 4287, 2005. *The Atom Syndication Format*. [Online], Available: http://www.ietf.org/rfc/rfc4287.txt [9 March 2010].

RFC 5023, 2007. *The Atom Publishing Protocol*. [Online], Available: http://www.ietf.org/rfc/rfc5023.txt [9 March 2010].

Rodriguez, P. et al, 2009. VaaS: Videoconference as a Service. *Proceedings of 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2009 (CollaborateCom 2009)*. Washington, DC, USA, pp. 1-11.

Thomas, D. and Heinemeier, D. 2006. *Agile Web Development with Rails: Second Ed*. The Pragmatic Bookshelf, USA.

Wasserman, S. and Faust, K., 1994. *Social network analysis: methods and applications*. Cambridge University Press, Cambridge, UK.